**Amendments to the Specification:**

Please amend the specification as follows:

Please substitute the following replacement paragraphs for the similarly-numbered paragraphs in the specification:

**[0020]** Both the client computer 104 and the server 102 contain, typically embedded in their operating systems (not shown), network protocol stacks 108 and 110. In the case of the Internet, these stacks would include TCP/IP stacks that are able to establish interconnections between two "sockets," one on the client computer 104 and one on the server 102. For web communication, socket number 80 is normally used, but a different socket is used in the case of secure communication and perhaps also in the case of communication between handheld devices that use [[a]] slightly different protocols in accordance with their special needs.

**[0021]** On the Internet, communication typically begins when some program entity within the client computer 104 wishes to send some form of message to the server 102, typically a request to have a web page downloaded from the server 102 and returned to the client computer 104. Such a request typically begins with "HTTP" followed by "://", which identifies the request as a "hypertext transfer protocol" request to retrieve a web page from a server 102. What follows next is the "Domain Name" of the server 102, such as "www.quert.com" ~~"www.abc.com"~~. This is typically followed by a "path" -- a subdirectory name string, such as "/main_directory/ ... /sub_directory/" -- and by the name of the web page document that is being requested for display, for example, "web_page.html". (The web page document may be a program, an ~~and~~ image, a file, or some other downloadable entity.)

**[0025]** If a true HTML web page has been requested, [[,]] the web page is simply found at 112 and is returned between the sockets 80 of the two TCP/IP stacks 110 and 108 over the Internet 106 and is displayed by a web browser 107 within the client computer 104. However, there are other possibilities. For example, a form of computer program written in a very high level interpretative language (a "CGI" script) may reside at the designated path and file name

address, in which case that program 112 is retrieved and is interpreted and executed by the operating system. Any extra data, such as cookie data, which was appended to the incoming message by the client computer, is passed to that program as operating system parameters that can control and affect the program's execution and that can be read into the program as incoming data. Such program might typically retrieve information from a database (not shown), assemble a customized HTML web page, and then return the web page to the browser 107 in the client computer 104 for display.

**[0027]** As can be seen, the browser 107 within the client computer 104, by assisting the user in formulating a proper query for the server 102, is thus able to trigger the execution of programs residing on the server 102, including servlets. The browser 107 typically displays a web page to the user that was downloaded from a server. The user, using a mouse (not shown) or other pointing device, is able to click upon URLs or "Universal Resource Locators" which are web addresses of the type illustrated in Figure 4 (but possibly lacking the "Cookie:..." suffix) residing within the web page. In response to the user clicking on such a URL in a web page, a URL request 116 (Figures 1 and 4) is generated in the format described above. As shown in Figure 4, the URL request includes information that is derived from cookies such as the cookies 122 and 200 residing on the client computer 104 that contain at least the suffix part or all of the "domain" name of the server 102 to which the URL request is directed and that optionally contain at least the prefix part or all of any designated directory path, as will be explained. Accordingly, once a cookie 118 is installed within the file system 120 of a client computer 104 at the request of a server 102, then any time the user clicks upon a page containing a URL that contains the address of that same server 102, typically all of the cookie information for the server 102 is sent over the Internet and back to the server 102 along with the request for the web page ~~having~~ that corresponds to the URL. Accordingly, if the web page request causes a servlet or interpretative program to be executed, the servlet or interpretative program receives, as part of the incoming message string, the names and the information contents of all the cookies placed into the client computer 104 by that particular server 102.

**[0029]** Figure 2 illustrates in more detail the actual data structure of a cookie. A typical cookie 200 includes a "domain" specification, such as "quert.com" ~~"abc.com"~~, which specifies the suffix portion or the entirety of the name of all servers 102 to which the cookie is to be returned whenever a request goes out to a server having the specified "domain" as the entirety or as the suffix portion of its Internet name. Thus, the two servers respectively named www.quert.com" and "www.xyz.quert.com" ~~"www.abc.com" and "www.xyz.abc.com"~~ would both be sent the contents of a cookie whose "domain" parameter was "www.quert.com" ~~"abc.com"~~ because that domain parameter matches the suffix portions of both of those server Internet names.

**[0036]** In Figure 4, the URL request 116 generated by the browser 107 requests that the server "www.abc.com" download the web page "web_page.html" to be found at directory and subdirectory path location "/main_directory/ ... /sub_directory/" using the transfer protocol "HTTP" (hypertext transfer protocol). This URL is indicated at 402 in Figure 4, and it identifies the server 102 by name and also the specific web page desired plus the directory and sub-directory path on the server 102 that leads to that web page. Appended to the URL 402 at 404, optionally (depending upon the presence of absence of cookies), is a string beginning with the HTTP command word "Cookie:" and followed by a series of one or more equality statements equating the name of a specific cookie with its value, and with multiple such statements, if present, separated by semicolons, as shown, if there is more than one. In this case, the assumption is that the cookie storage area 118 of the file system 120 contains two cookies 122 and 200 which each contain the domain name suffix "quert.com" ~~"abc.com"~~, one containing the name string "XYZ" and another containing the name string "PDQ" as is shown in Figure 1. Accordingly, the "Cookie:" command 404 includes two strings, each containing an equal sign, a name, and a value, and separated by a semicolon which are sent along to the server 102 along with the "HTTP://" command prefix and the URL specified at 402 in Figure 4.

**[0045]** At step 608, the HTTP header for this HTML page contains, for each bad cookie, a "Set Cookie" command, in the format indicated in Figure 3. [[.]] Each such command

contains a bad cookie's name for its data value, plus a blank value for the cookie; a blank path value, a domain equal to the server 102's name suffix or full name, and also containing the expiration code set to cause the cookie to expire immediately. This HTML page may also include a simple message stating that the bad cookies have been deleted and, optionally, repeating their names and values for the user. This document is then transmitted back to the browser 107 which, in receiving theses new cookies, automatically erases or neutralizes the previous cookies containing the bad values and replaces them with cookies that expire immediately and that contain blank data values.